Movement as higher-order structure building

Patrick D. Elliott (ZAS)

June 11, 2019

Universität Göttingen

- Current theories of movement at give rise to conceptual worries *vis a vis* interface requirements. Is *Internal Merge* causing more problems than it solves?
- The goal here: develop a radically different perspective on syntactic displacement as *higher-order structure building*, borrowing well-established standard mechanisms from Montagovian semantics for dealing with semantic displacement (i.e., *scope*).
- Some payoffs include:
    - No need for *trace conversion*.
    - An account of Müller's (2001) *generalized order preservation*.
    - An account of the interaction between scrambling and scope-taking in scope-rigid languages such as Japanese.

- A (non-standard) overview of *movement* in minimalist syntax + some conceptual worries.
- An analogy between overt *syntactic* displacement and the QR-analysis of *semantic* displacement.
- Reifying the analogy in a purely derivational system via *higher-order structure building*.
- An analysis of *wh*-movement.
- An extension to quantifier raising and scrambling.
- Finish!

Formal Preliminaries

- Since this is a theory talk, let's try to be precise about the operations we're using.
- *Types* will help us give an explicit treatment of syntactic operations as *functions*.
- Fortunately, we're only going to need one primitive type: Let $\mathbb{t}$ be the *type* of a Syntactic Object (SO). Whenever I talk about syntactic types or variables over SOs, I'll use $\mathbb{blackboard}$ font.

- We can't really do anything interesting with just our primitive type t. We'll also avail ourselves of *function types*.
- I'll use (→) as the constructor for *function types* (cf., e.g., Heim & Kratzer 1998 who use ⟨.⟩).
- $a \to b$ is the type of a function from things of type a to things of type b.
- Where Heim & Kratzer write $\langle\langle e, t\rangle, t\rangle$, i'll write $(e \to t) \to t$.
- N.b. that (→) is *right-associative*, so $e \to e \to t \equiv e \to (e \to t)$.

- We'll take as our starting point the hypothesis that the basic structure-building operation in natural language is MERGE (Chomsky 1995).
- We define MERGE in a pretty standard way – it's a *function* that takes two SOs, and returns a new (unlabelled) SO.

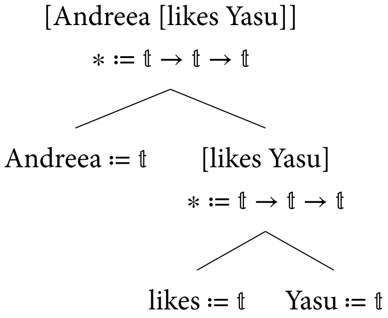
(1) MERGE (def.)
$\mathbb{X} * \mathbb{Y} \coloneqq [\mathbb{X}\ \mathbb{Y}]$ $\quad ::= \mathbb{t} \rightarrow \mathbb{t} \rightarrow \mathbb{t}$

- NOTE: following, e.g., Stabler (1997), we assume that merge is asymmetric:

$$\mathbb{X} * \mathbb{Y} \neq \mathbb{Y} * \mathbb{X}$$

- Merge successively applies to sos constructing a structured representation, as in (2):

(2)



- Important: the tree is a graph of the *derivation*, rather than a representation in its own right.

- Let the type of the atomic unit of syntactic computation (a lexical object, root, etc.), be L. This allows us to define $\mathbb{t}$ recursively.

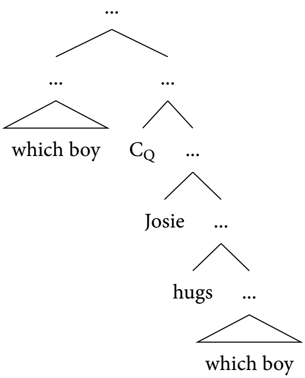
$$\mathbb{t} := \mathsf{L} \mid [\mathbb{t}]$$

Movement in MERGE-based frameworks

- Certain expressions (such as *wh*-expressions) are pronounced in positions other than where they're interpreted – or, more precisely, where a *part* of their meaning (the variable) is interpreted.
- The standard approach to this phenomenon in minimalism is Internal Merge.
- This can be cashed out in two different ways: the *copy theory* and the *multidominance theory* of movement.
- I'll just present the copy theory for exposition, but multidominance approaches are subject to the same issues.

- According to the copy theory, movement involves merging a *copy* of an SO contained within the derived syntactic structure.

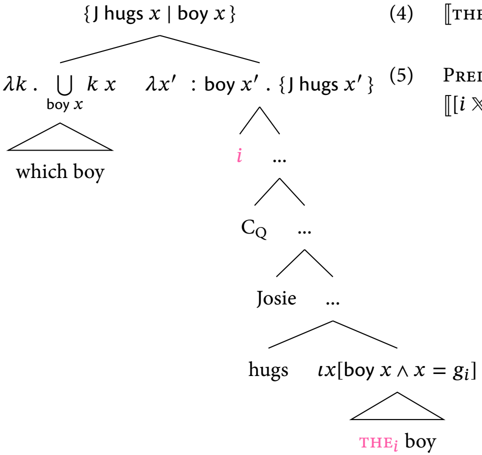
(3)



- It's not trivial to implement Internal Merge as a function. It should traverse through the constructed syntactic representation for the SO to be copied-and-remerged (although see Collins & Stabler 2016 for a local formulation).

- Regardless of how Internal Merge is implemented, the representation interpreted by the semantic component must look something like this (Fox 2002, Sauerland 2004):

TRACE CONVERSION II



(4) $[\![\textsc{the}_i]\!]^g = \lambda P \,.\, \iota x[P\ x \wedge x = g_i]$

(5) PREDICATE ABSTRACTION (def.)
$[\![[i\ \mathbb{X}]]\!]^g = \lambda x \,.\, [\![\mathbb{X}]\!]^{g[i \to x]}$

- How do we get from a copy-theoretic representation to the representation required by the semantics?
- First off, we need a syntactic operation that applies to the lower copy, and replaces the determiner with THE_i.

(6) Trace Conversion (def.)
$\text{TC}\ [\mathbb{D}\ \mathbb{N}]_i \coloneqq [\text{THE}_i\ \mathbb{N}]$

- We also need a syntactic operation that places a *binding index* immediately below the higher copy, in order to trigger abstraction over the lower copy.

- Due to the demands of the interface, much of the conceptual appeal of Internal Merge is lost.
- Trace Conversion = the name for a problem, rather than a solution (although, see Fox & Johnson 2016 for a more principled account).
- Goal for the next section: an approach which retains the conceptual appeal of Internal Merge, where meaning-computation can proceed in tandem with movement derivations, without the need for syntactic magic, such as Trace Conversion, and binding index insertion.

Higher-order structure building

- Exploring a (failed?) analogy with between displacement as *Quantifier Raising*.
- Reifying the analogy in a derivational framework.
- Introducing our players:

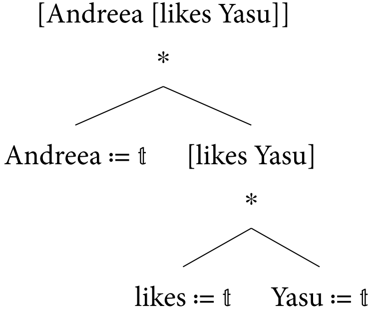
scopal-Merge ($\star$) Our version of *internal merge*.

Lift ($\uparrow$) Converting an SO into a trivial scope-taker.

Higher Order Merge ($\circledast$) A combinatorics for *scopal* syntactic values.

- Before we present our analysis, let's entertain an analogy.
- Imagine that derivation graphs are, themselves, fully-fledged representations.

(7)



- Now, let's define a new unary operation, s-Merge (i.e., *scopal merge*), which we'll write as $(\star)$. It's just defined in terms of merge + lambdas and variables.

(8) $\star\, \mathbb{X} \coloneqq \lambda k\,.\, \mathbb{X} * (k\, \mathbb{X})$ $\qquad (\star) ::= \mathbb{t} \to (\mathbb{t} \to \mathbb{t}) \to \mathbb{t}$

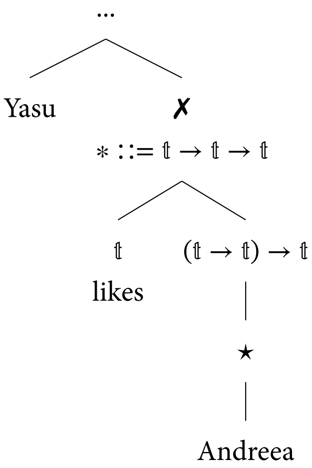
- $(\star)$ takes a so, and shifts it into a function that takes a function from sos to sos, and returns an so.

(9) $\star$ Andreea $= \lambda k\,.$ Andreea $*$ (k Andreea) $\qquad (\mathbb{t} \to \mathbb{t}) \to \mathbb{t}$

- You can think of $\star$ as a function from an so to something that *takes scope* over sos.

An analogy with QR iii

- If we apply ($\star$) to an SO over the course of our derivation, we end up with a type mismatch. MERGE takes two arguments of type $\mathbb{t}$.



- In order to resolve this type mismatch let's assume we can scope out the $\star$–shifted SO via QR – assuming that something of type $(\mathbb{t} \to \mathbb{t}) \to \mathbb{t}$ binds a type $\mathbb{t}$ variable.
- The result will be a kind of *derivational scope*; ($\star$ Andreea) contributes the SO Andreea locally, and the function ($\lambda\mathbb{X}$. Andreea $*$ $\mathbb{X}$) takes scope.
- When we compute the result, we will end up with a copy-theoretic representation.

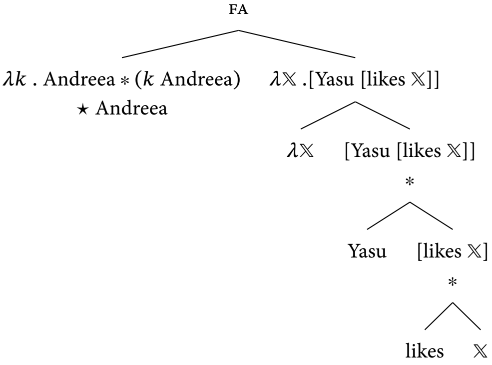
An analogy with QR V

$$[\lambda k\,.\ \text{Andreea} * (k\ \text{Andreea})]\ (\lambda \mathbb{X}\,.[\text{Yasu [likes } \mathbb{X}]])$$

$$= \text{Andreea} * ([\lambda \mathbb{X}\,.[\text{Yasu [likes } \mathbb{X}]]]\ \text{Andreea})$$

$$= \text{Andreea} * ([\text{Yasu [likes Andreea]]})$$

$$= [\text{Andreea [Yasu [likes Andreea]]]}$$



- Unfortunately, the analogy with QR breaks down – since derivation graphs are not themselves *representations*, it doesn't really make sense conceptually to posit an operation of QR that applies to a *derivation graph*.
- What we want, intuitively, is a way of compositionally integrating scopal values into a computation.
- We'll model our approach on Barker & Shan's (2014) *continuation semantics*.

- *Scopal* values are of type $(\mathsf{a} \to \mathsf{b}) \to \mathsf{b}$. Barker & Shan (2014) introduce a convenient notational shortcut for scopal types – tower types.

(10) $$\frac{\mathsf{b}}{\mathsf{a}} := (\mathsf{a} \to \mathsf{b}) \to \mathsf{b}$$

- Similarly, scopal values themselves can be rewritten using tower notation:

(11) $$\frac{f\,[]}{x} := \lambda k\,.\,f\ (k\ x)$$

- Standard entries for quantificational expressions can be rewritten using tower notation, like so:

(12) $\text{everyone} = \lambda k \,.\, \forall x[k\ x] \quad ::= (\mathsf{e} \to \mathsf{t}) \to \mathsf{t}$

$$= \frac{\forall x[]}{x} \qquad ::= \frac{\mathsf{t}}{\mathsf{e}}$$

- We can now rewrite the syntactic operation s-Merge (⋆) using tower notation:

$$(13) \quad \star\, \mathbb{X} \coloneqq \frac{\mathbb{X} * []}{\mathbb{X}} \qquad\qquad (\star) ::= \frac{\mathfrak{t}}{\mathfrak{t}}$$

- Recall that ⋆–shifting a so gives rise to a type mismatch in the derivation. Let's explore a different way of incorporating ⋆–shifted sos into the derivation.

- In order to do this, we need to define two new derivational operations.
- Lift takes an so and returns a *trivially* scopal/higher-order so.

(14) Lift (def.)

$\mathbb{X}^{\uparrow} \coloneqq \lambda k \,.\, k\ \mathbb{X}$ $\qquad (\uparrow) :: = \mathbb{t} \to \frac{\mathbb{t}}{\mathbb{t}}$

- Higher Order Merge provides us with a way of merging two higher-order/scopal syntactic objects.

(15) Higher Order Merge (def.)

$$m \circledast n$$
$$:= \lambda k \,.\, m\, (\lambda \mathbb{X} \,.\, \lambda n \,.\, (\lambda \mathbb{Y} \,.\, \lambda k \,.\, (\mathbb{X} * \mathbb{Y})))$$

$$(\circledast) ::= \frac{\mathbb{t}}{\mathbb{t}} \to \frac{\mathbb{t}}{\mathbb{t}} \to \frac{\mathbb{t}}{\mathbb{t}}$$

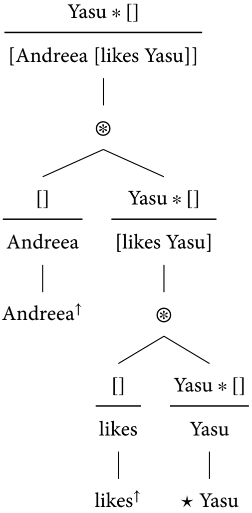
- In order to see what's going on, it will be easier to rewrite these functions using tower notation.
- Lift coverts an so into a trivial tower.

$$\mathbb{X}^{\uparrow} := \frac{[]}{\mathbb{X}}$$

- HO Merge provides a way of *merging* two towers.

$$\frac{f\ []}{\mathbb{x}} \circledast \frac{g\ []}{\mathbb{Y}} := \frac{f\ [g\ []]}{\mathbb{X} * \mathbb{Y}}$$

- Now we have everything we need to incorporate ⋆−shifted sos into the syntactic derivation:



- Finally, we need a syntactic operation to *lower* a higher-order so back down to an ordinary so. We can define Lower simply as the identity function.

(16) Lower (def.)

$$\downarrow m \coloneqq m\ \mathsf{id} \qquad\qquad (\downarrow) :: = \frac{\mathbb{t}}{\mathbb{t}} \to \mathbb{t}$$

- *Lowering* the higher-order so gives us the same result as the copy theory of movement!

$$\downarrow \left(\frac{\text{Yasu} * [\,]}{[\text{Andreea [likes Yasu]}]}\right) = [\text{Yasu [Andreea [likes Andreea]]}]$$

- Since we're adopting a radically derivational perspective, we don't really *need* to refer to the outputted structural representations for anything. Let's simplify things and just treat MERGE as *concatenation* (see Kobele 2006 for a thorough demonstration that this is harmless).

(17) $\mathbb{X} * \mathbb{Y} \coloneqq \mathbb{X} : \mathbb{Y}$

- On this view, it's natural to redefine $\star$ such that the local value has null phonological content:

(18) $\star\, \mathbb{X} \coloneqq \dfrac{\mathbb{X} * []}{\varnothing}$

(19) Yasu, Andreea likes.

(20) $((\text{Andreea}^{\uparrow}) \circledast ((\text{likes}^{\uparrow}) \circledast (\star\text{Yasu})))^{\downarrow}$
$= (\lambda k\,.\text{Yasu} * (k\ (\text{Andreea} : \text{likes} : \varnothing)))^{\downarrow}$
$= \text{Yasu} : \text{Andreea} : \text{likes} : \varnothing$

Extension to wh-movement

- In order to extend the proposal to *wh*-movement, we must make it more syntactically realistic. We'll treat SOS as *feature bundles*; MERGE concatenates feature bundles.
- We can now redefine merge as a *feature sensitive* operation.
- Merging an SO $\mathbb{X}$ with an uninterpretable *Q* feature with another SO ($\mathbb{Y} : \mathbb{Z}$) results in *ungrammaticality* (♯), unless the head $\mathbb{Y}$ carries an interpretable *Q* feature.

(21) a. $\mathbb{X}_{[uQ]} * (\mathbb{Y}_{[iQ]} : \mathbb{Z}) = \mathbb{X} : \mathbb{Y}_{[iQ]} : \mathbb{Z}$

b. $\mathbb{X}_{[uQ]} * \mathbb{Y} = \sharp$

c. $\mathbb{X} * \mathbb{Y} = \mathbb{X} : \mathbb{Y}$

- This needs to be generalised, but this will do for now.

- We can now additionally redefine (⋆) in a feature sensitive way:

(22) $\star\ \mathbb{X}_{[d,uQ]} \coloneqq \dfrac{\mathbb{X}_{[d,uQ]}}{\varnothing_{[d]}}$

- We now have everything we need to account for feature-driven movement in a more realistic way:

(23) a. $((\text{C}^{\uparrow}_{[iQ]}) \circledast ((\text{Andreea}^{\uparrow}) \circledast ((\text{likes}^{\uparrow}) \circledast (\star\text{who}_{[d,uQ]}))))^{\downarrow}$

b. $(\lambda k\ .\ \text{who}_{[d,uQ]} * k\ (\text{C}_{[iQ]} : \text{Andreea} : \text{likes} : \varnothing_{[d]}))^{\downarrow}$

c. $\text{who}_{[d]} : \text{C}_{[iQ]} : \text{Andreea} : \text{likes} : \varnothing_{[d]}$

- *order preservation effects* are pervasive in syntax (Müller 2001), e.g., *superiority effects* in English and multiple *wh*-fronting languages such as Bulgarian.

(24) a. I wonder who^x t_x bought what^y.

b. *I wonder what^y who bought t_y.

(25) a. *Koj kakvo kupuva?*
Who what buys?

b. **Kakvo koj kupuva?*
What who buys?

- Order-preservation falls out as the unmarked case in the system outlined here. This is because HO Merge (repeated below) sequences movements from left-to-right.

$$\frac{f\ []}{\mathbb{x}} \circledast \frac{g\ []}{\mathbb{Y}} := \frac{f\ [g\ []]}{\mathbb{X} * \mathbb{Y}}$$

- We ignore the feature calculus here for ease of exposition:

(26) a. $\downarrow\ (((\star\ \text{who}) \circledast ((\text{buys}^{\uparrow}) \circledast (\star\ \text{what}))))$

b. $=\downarrow\ (\lambda k\ .\ \text{who} * (\text{what} * (k\ (\varnothing : \text{buys} : \varnothing))))$

c. $= \text{who} : \text{what} : \varnothing : \text{buys} : \varnothing$

- In this system, semantic computation can proceed *in tandem* with syntactic computation. We'll assign a single meaning to a *wh*-expression which will predict that it scopes exactly at the position it's moved to.
- We adopt a generalized Karttunen semantics for *wh*-expressions – they scope over question meanings and return question meanings (Cresti 1995, Charlow 2014, Elliott 2017)

(27) $[\![\text{who}]\!] \coloneqq \lambda k\,.\ \bigcup_{\text{person } x} k\ x$ $\qquad (e \to \{t\}) \to \{t\}$

- Note that *wh*-expressions have a *scopal* semantics – we can scope them using semantic correlates of LIFT and HO MERGE (Barker & Shan 2014).

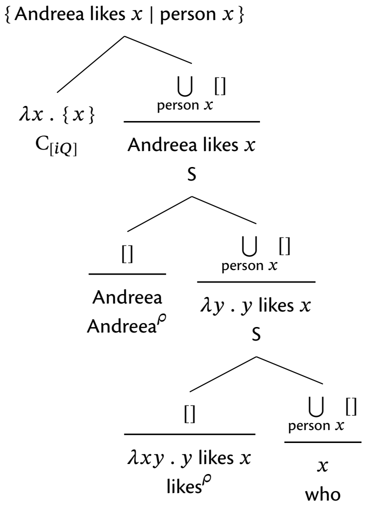
- We take the semantic correlate of the syntactic operation LIFT to be RETURN (ρ).

(28) $x^{\rho} := \dfrac{[]}{x}$ $\quad$ $(\rho) ::= \mathsf{a} \to (\mathsf{a} \to \{\, \mathsf{b} \,\}) \to \{\, \mathsf{b} \,\}$

- We take the semantic correlate of the syntactic operation HO MERGE to be SCOPAL FUNCTION APPLICATION (S).

(29) $\dfrac{f\ []}{x}\ \mathsf{S}\ \dfrac{g\ []}{y} := \dfrac{f\ [g\ []]}{\mathsf{A}\ x\ y}$

- Finally, we take the meaning of $C_{[iQ]}$ to be singleton-set formation.



- Note the isomorphism between the semantic computation and syntactic computation. Both are computed step-by-step, in tandem.

(30) a. Syntax:
$[\![C_{[iQ]}]\!]\ (([\![\text{Andreea}]\!]^{\rho})\ \mathsf{S}\ (([\![\text{likes}]\!]^{\rho})\ \mathsf{S}\ [\![\star\ \text{who}_{[d,uQ]}]\!]))$

b. Semantics:
$((C^{\uparrow}_{[iQ]}) \circledast ((\text{Andreea}^{\uparrow}) \circledast ((\text{likes}^{\uparrow}) \circledast (\star\text{who}_{[d,uQ]}))))$

- There is no need for anything like trace conversion. In the syntax, movement corresponds to scoping the features + phonological content of a syntactic object, in the semantic component, anything with a scopal semantics exhibits *interpretive* displacement via the same mechanisms.

- Merge in the syntax corresponds to Function Application in the semantics: $(*) \approx \mathsf{A}$
- When a moved expression is *scopal* (i.e. interpreted in its derived position):
  - Lift in the syntax corresponds to Return in the semantics (in fact, they're polymorphic instantiations of the same function): $(\uparrow) \approx (\rho)$
  - HO Merge in the syntax corresponds to Scopal Function Application in the semantics: $(\circledast) \approx \mathsf{S}$

- In this system, quantifier raising simply involves a scopal semantics with a non-movement syntax. There is in fact no need for covert movement.

(31) a. Syntax:
some linguist $*$ (hates $*$ [every philosopher])
$= [\text{some linguist}] : \text{hates} : [\text{every philosopher}]$

b. Semantics:

$$\left(\frac{\exists y[\text{linguist}\ y \wedge [\,]]}{y}\ \mathsf{S}\ \left(\frac{[\,]}{\text{hates}}\ \mathsf{S}\ \frac{\forall x[\text{phil}\ x \to [\,]]}{x}\right)\right)^{\downarrow}$$

$= \exists y[\text{linguist}\ y \wedge \forall x[\text{phil}\ x \to y\ \text{hates}\ x]]$

- Note that the unmarked case in this system is *surface scope*. This is a good prediction for scope-rigid languages like German, but we need to do a little more to get inverse scope.

- Barker & Shan (2014) show that we can derive inverse scope by *internally lifting* (⇈) the lower quantifier, and (re-)lifting the higher quantifier. Details suppressed here but see Barker & Shan.

$$\left(\frac{[]}{\dfrac{\exists y[\text{ling } y \wedge []]}{y}} \;\; \mathsf{S} \left(\frac{[]}{\dfrac{[]}{\text{hates}}} \;\; \mathsf{S} \;\; \frac{\forall x[\text{phil } x \to []]}{\dfrac{\_\_}{x}} \right) \right)^{\Downarrow}$$

$$= \left(\frac{\forall x[\text{phil } x \to []]}{\dfrac{\exists y[\text{ling } y \wedge []]}{y \text{ hates } x}} \right)^{\Downarrow} = \forall x[\text{phil } x \to \exists y[\text{ling } y \wedge y \text{ hates } x]]$$

- Let's assume that internal lift is freely available in English, without any syntactic reflex. This predicts the availability of scopal ambiguities.
- Languages such as Japanese and Hindi are ordinarily scope rigid however; scopal ambiguities may arise if a scopal expression is *scrambled*.

(32) a. *Dareka-ga* *daremo-o* *sonkeisiteiru*
someone-NOM everyone-ACC admire

some > every, *every > some

b. *daremo-o* *dareka-ga* *t* *sonkeisiteiru*
everyone-ACC someone-NOM *t* admire

some > every, every > some

- There's a very natural perspective to adopt in languages such as Japanese and Hindi – *internal lift* isn't freely available, rather, it is the semantic reflex of s-Merge ($\star$).

(33) *Syntax*:
$([\text{Some philosopher}]^{\uparrow} \circledast ((\text{hates}^{\uparrow})\ (\star\ [\text{every linguist}])))^{\downarrow}$
$= [\text{every linguist}] : [\text{some philosopher}] : [\text{hates}] : \varnothing$

(34) *Semantics*:
$((\llbracket\text{some philosopher}\rrbracket^{\rho})\ \mathsf{S}\ ((\llbracket\text{hates}\rrbracket^{\rho\circ\rho})\ \mathsf{S}\ (\llbracket\text{every linguist}\rrbracket^{\uparrow\uparrow})))^{\downarrow\downarrow}$

$$= \left(\frac{\llbracket\text{every linguist}\rrbracket\ \ \lambda x\ []}{\dfrac{\llbracket\text{some philosopher}\rrbracket\ \ \lambda y\ []}{y \text{ hates } x}} \right)^{\downarrow\downarrow}$$

$= \forall x[\text{ling } x \rightarrow \exists y[\text{phil } y \wedge y \text{ hates } x]]$

- But scrambling doesn't just give rise to inverse scope – it gives rise to *scopal ambiguities*
- We can account for this by simply positing an *implicational* rather than a *one-to-one* relationship between (⇈) and (⋆) – (⇈) (in Japanese) implies (⋆) in the syntactic computation, but not vice versa.
- In other words, (⇈) is an *optional* semantic reflex of (⋆), but it is not permitted in the absence of (⇈).

Conclusion

- How to account for the following within this framework:
  - *locality* – has a natural treatment in terms of obligatory lowering; see Charlow (2014) on scope islands.
  - *Successive-cyclicity* – has a natural treatment in terms of *lowering* followed by re-s-MERGEing.
  - *Reconstruction* – see Barker & Shan (2014) for a detailed treatment consistent with this system.
  - *Late merge* – more difficult, but can be analyzed without copies once more sophisticated mechanisms for scope-taking (*indexed continuations*) are adopted. I'll come back to this in future work.

- In this talk, I've suggested that we can take a cue from the formal semantics literature, and treat syntactic displacement as a kind of *syntactic* scope-taking.
- This move has a major conceptual advantage – semantic computation can proceed *in tandem with* syntactic computation. There is no need for any *ad-hoc* mechanism for interpreting movement.
- We've mentioned a couple of interesting empirical payoffs – the analysis of generalised order preservation, and scrambling.
- A more thorough exploration of the properties of this system will have to wait for another time!

Thanks for listening!

References I

Barker, Chris & Chung-chieh Shan. 2014. *Continuations and natural language*. (Oxford studies in theoretical linguistics 53). Oxford University Press. 228 pp.

Charlow, Simon. 2014. *On the semantics of exceptional scope*.

Chomsky, Noam. 1995. *The minimalist program*. (Current Studies in Linguistics 28). Cambridge Massachussetts: The MIT Press. 420 pp.

Collins, Chris & Edward Stabler. 2016. A Formalization of Minimalist Syntax. *Syntax* 19(1). 43–78.

Cresti, Diana. 1995. Extraction and reconstruction. *Natural Language Semantics* 3(1). 79–122.

Elliott, Patrick D. 2017. Nesting habits of flightless *wh*-phrases. unpublished manuscript. University College London.

Fox, Danny. 2002. Antecedent-contained deletion and the copy theory of movement. *Linguistic Inquiry* 33(1). 63–96.

Fox, Danny & Kyle Johnson. 2016. QR is restrictor sharing. In Kyeong-min Kim et al. (eds.), *Proceedings of the 33rd West Coast Conference on Formal Linguistics*, 1–16. Somerville, MA: Cascadilla Proceedings Project.

Heim, Irene & Angelika Kratzer. 1998. *Semantics in generative grammar*. (Blackwell textbooks in linguistics 13). Malden, MA: Blackwell. 324 pp.

Kobele, Gregory. 2006. *Generating copies - An investigation into structural identity in language and grammar*. UCLA dissertation.

Müller, Gereon. 2001. Order Preservation, Parallel Movement, and the Emergence of the Unmarked. In.

Sauerland, Uli. 2004. The interpretation of traces. *Natural Language Semantics* 12(1). 63–127.

Stabler, Edward. 1997. Derivational Minimalism. In, 68–95.